EXHIBIT 45

UNITED STATES PATENT AND TRADEMARK OFFICE

_____

BEFORE THE PATENT TRIAL AND APPEAL BOARD

_____

ARISTA NETWORKS, INC.
Petitioner

v.

CISCO SYSTEMS, INC.
Patent Owner

_____

Case IPR2016-00119
Patent 7,047,526

_____

PATENT OWNER PRELIMINARY RESPONSE

Mail Stop "PATENT BOARD"
Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

IPR2016-00119
U.S. Pat. No. 7,047,526

TABLE OF AUTHORITIES

IPR2016-00119
U.S. Pat. No. 7,047,526

EXHIBIT LIST

Exhibit Number	Document Description
2001	Manuel Rubio-Sanchez, Jamie Urquiza-Fuentes, and Cristobol Pareja-Florea, *A Gentle Introduction to Mutual Recursion* (2008)

I.   Introduction.

The Cisco '526 patent provides a system that allows system administrators to use a single tool to manage and execute multiple administration and diagnostic programs using a novel and unique command parse tree. The use of this single management tool overcame significant challenges in the prior art because previous management systems required separate command languages, each with their own unique function and syntax. The '526 patent brought management order to this chaos.

The Petition presents a single ground of rejection, relying solely on Martinez-Guerra to render obvious claims 1–26 of the '526 patent. But, Martinez-Guerra does not teach or suggest "management programs" or the structure of the command parse tree recited in each of the independent claims. First, Martinez-Guerra does not disclose the claimed "management programs" because the tools disclosed by Martinez-Guerra are not separate or external from Martinez-Guerra's parser-translator component. Second, each independent claim recites a command parse tree that performs validation and specifies a presubscribed command that has elements "each specifying at least one corresponding generic command component and a corresponding at least one command action value." Petitioner argues that Martinez-Guerra's phrase structure rules, which are merely used for validation, are the claimed command parse tree. But, Martinez-Guerra's phrase structure rules

- 1 -

cannot be the claimed "command parse tree" because they do not contain

"elements each specifying a command action value" as each independent claim

requires. Therefore, Petitioner's sole challenge fails for all claims and the Petition

should be denied.

Petitioner's sole challenge to dependent claims 2, 3, 11, 12, 15, 16, and 25

fails for additional reasons. Martinez-Guerra does not teach or suggest the claimed

command word translation table (claims 2, 11, and 15) or recursively traversing a

command parse tree (claims 3, 12, 16, and 25). And, for these additional reasons,

the Petition should be denied for dependent claims 2, 3, 11, 12, 15, 16, and 25.

II.   The '526 patent presents a novel approach for command and interface control of Operating Administration and Monitoring tools.

A.   OAM tools facilitate the operation of computer systems.

Operating Administration and Monitoring (OAM) tools enable the operation,

administration, management, and maintenance of a computer system. These OAM

tools are "resources used as administration and/or diagnostic tools for complex

processor-based executable software systems, such as software-based unified

messaging software systems." (Ex. 1001, '526 patent, 1:11–15.) OAM tools include

Real Time Monitoring (RTM) programs that allow administrators to monitor and

control various states and processes within the computer system. ('526 patent,

1:15–18.)

Specifically, administrators can review, evaluate and modify the overall processes and functions performed on data in real-time. For example, a RTM program may "generate a real-time display (i.e., 'a screen') of selected parameters during execution of a prescribed process" or "provide a diagnostic resource that enables resetting of various states or variables within the prescribed process." ('526 patent, 1:18–23.) Other examples of OAM tools include "external binary files that execute in response to a procedure call, and Simple Network Management Protocol (SNMP) agents or scripts configured for generating an e-mail message as an alarm in response to a detected event." ('526 patent, 1:23–27.)

B. Using multiple system administration and diagnostic tools presented problems and complexities for system administrators.

As computer systems and networks became more complex, system administrators often used multiple OAM tools within the system to increase the available administration and diagnostic tools for improved system performance. ('526 patent, 1:28–31.) However, the use of multiple OAM tools presented the system administrator with various disadvantages because each tool had its own distinct functions and syntax. For example, as the number of different management tools increased, the use of multiple RTM programs and other OAM tools created a significant burden on system administrators who were required to have knowledge

of the different function names and syntax formats of numerous commands. ('526

patent, 1:31–37.)

C.   The '526 patent's unified command language enables the use of generic commands to execute the OAM tools.

The '526 patent addresses the significant problems identified in the previous

section by enabling a user to execute multiple management programs and their

respective command and control functionality using a single tool. ('526 patent,

1:41–44.) This solution provides system administrators with a simple command

language that controls multiple administrative tools without requiring the system

administrator to learn the specific functions, command formats, and syntax for

each OAM tool. ('526 patent, Abstract.) Specifically, the '526 patent provides a

unified administration and diagnostic tool that "incorporates the functionality of all

external administrative executable binary files, RTM programs, agent manipulation

scripts, and various requested snapshot queries, as well as including an extensive

help system." ('526 patent, 3:21–27.)

The '526 patent provides system administrators with a single set of

commands and syntax for multiple administrative and maintenance tools. Figure 1

(reproduced below) illustrates an embodiment of the '526 patent system.
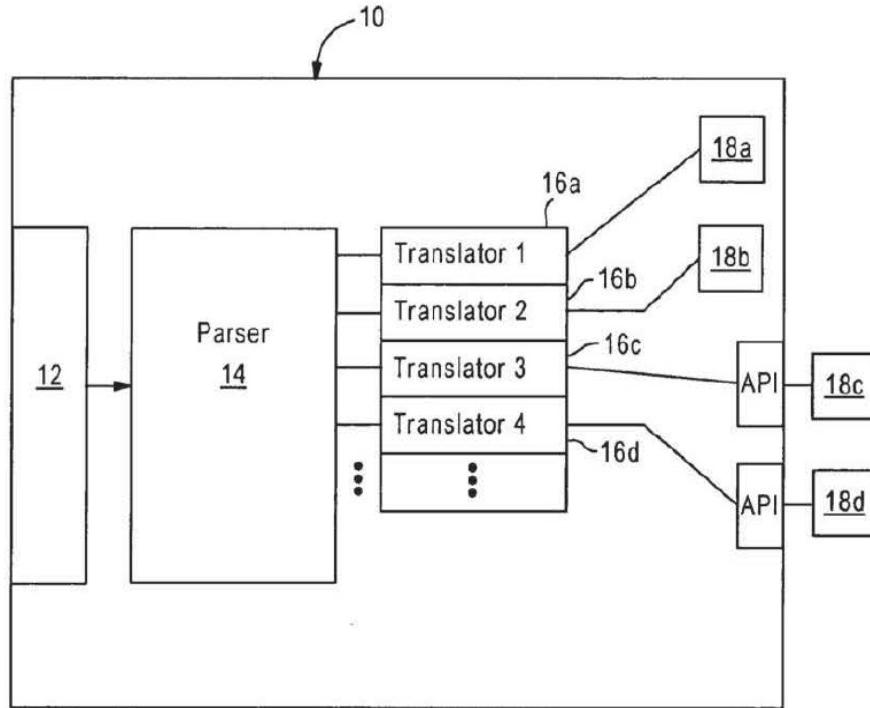
Figure 1

The system of Figure 1 includes a user input interface 12, a parser 14, a

plurality of translators 16 and a plurality of management tools 18 (*i.e.*, OAM

tools). When a system administrator enters a generic command via interface 12,

parser 14 validates the generic command. ('526 patent, 2:60–65.) Thereafter,

translators 16 issue commands to respective management programs 18 according

to respective command formats. ('526 patent, 2:60–3:1.) The patent explains that

parser 14 and the translators 16 "provide [system administrators with] a generic

command syntax that integrates the functionality of the different [management]

tools[,] and that automatically select[]the appropriate command for the best tool for

executing a given generic command." (*See* '526 patent, 3:27–31.) The system

administrator issues commands based on the relative functions, as opposed to the specific syntax for a corresponding OAM tool, resulting in a single point of entry for administering and maintaining complex software based systems. ('526 patent, 4:54–60; 3:31–35.)

D.   The '526 patent command parse tree.

The '526 patent uses a novel command parse tree to achieve its administration and diagnostic tool. The command parse tree includes multiple elements, and each element includes at least one corresponding generic command component and at least one corresponding command action value. ('526 patent, 1:51–54.)

A generic command may have one or more input command words. The parser validates a received generic command by comparing each input command word to elements of the command parse tree. ('526 patent, 1:48–51.) The search of the command parse tree occurs by recursively traversing the tree and determining, for the received generic command, a tree element that best matches the generic command. ('526 patent, 3:51–61.) Once the system identifies the tree element best matching the generic command, the selected management program issues a prescribed command based on the command action value of the identified tree element. ('526 patent, 1:54–58.)

Figure 2 (reproduced below with annotations) illustrates in detail an embodiment of the '526 patent parser which includes the command parse tree 22 and a command word translation table 20.
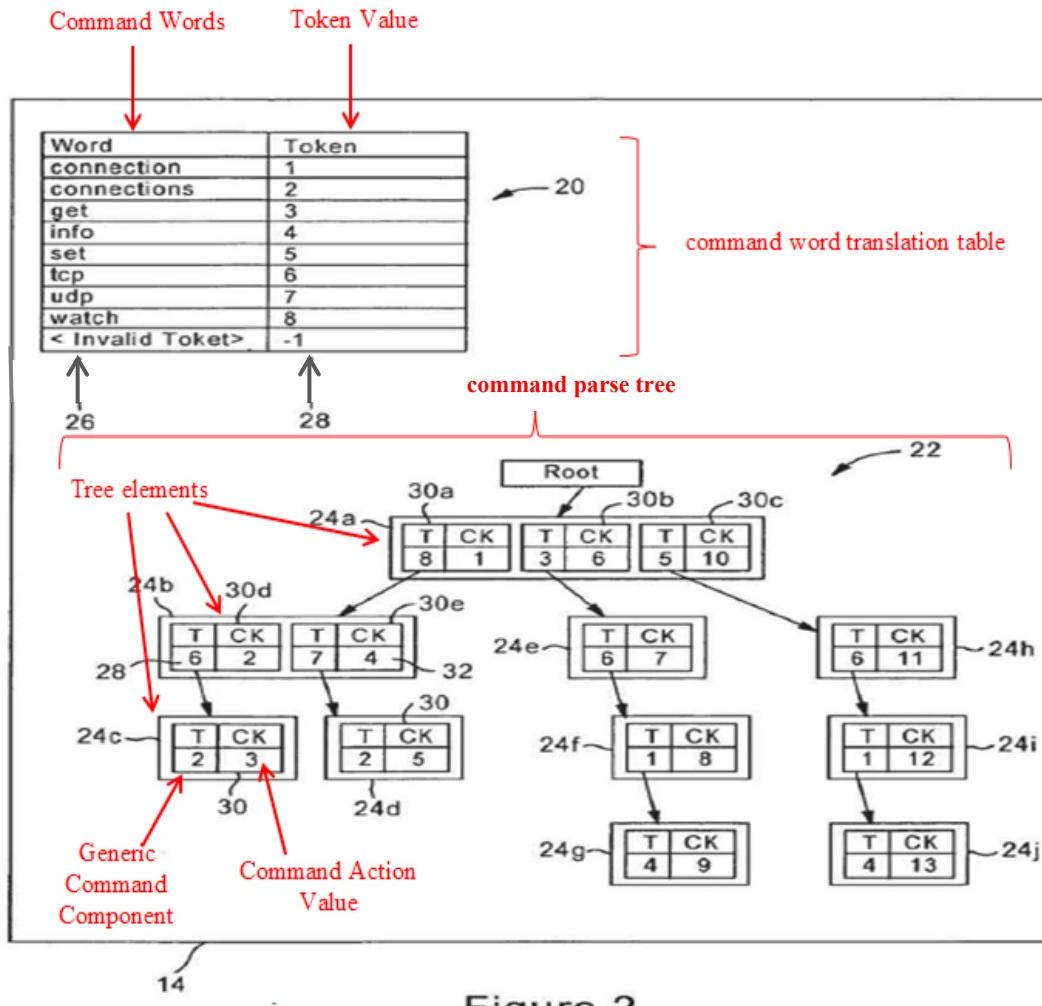
Figure 2

The command word translation table 20 stores the command words 26 that are valid according to the generic syntax. ('526 patent, 3:43–45.) Additionally, for each valid command word, the command word translation table also stores a corresponding token value 28. Parser 14 recursively traverses command parse tree

22 to identify tree elements that match a given generic command word. ('526 patent, 4:3–12.) The '526 patent parser validates a received generic command by first locating a command word and its corresponding token in the command word translation table. Parser 14 then compares the command word token to the command parse tree 22 to identify the best match. ('526 patent, 3:46–51.)

Each tree element 24 of the command parse tree includes at least one token-command key pair 30 that specifies a token (T) 28 and a corresponding command key (CK) 32. ('526 patent, 3:51–53.) The parser uses the token-command key pair to identify the appropriate prescribed command for a management program. ('526 patent, 3:53–55.) If the parser determines that only a portion of the received generic command is valid (*e.g.*, only the first three command words are valid from a total of 4 command words received), the parser selects the command key 32 for the matching token 28 from the last valid tree element 24 (*e.g.*, the tree element matching the third valid command word). ('526 patent, 3:58–61.) Thus, even if the entire received generic command is not valid, a generic command still may be executed for a particular management program. (*See* '526 patent, 4:37–51.)

While other parsing and translating systems, such as the one disclosed by Martinez-Guerra aimed to solve a similar problem, such systems addressed the problem using a different technique—a technique that did not use a command parse tree having elements where *each* element *specified a generic command*

component and at least one corresponding command action value. As a result,

systems such as Martinez-Guerra could not execute a generic command for a

partially valid input—a novel aspect accomplished by the '526 patent, because of

the unique command parse tree and its structure. (*See* '526 patent, 4:37–51; 3:58–

61.)

III.   Claim construction.

A.   Petitioner adopted Patent Owner's construction from the co-pending litigation for several claim terms.

Petitioner proposed constructions for four claim terms—"management

program," "command parse tree," "recursively traversing," and "means for

validating"/"validating means." (Petition, pp. 12–14.) For the terms "management

program" and "command parse tree," Petitioner has adopted the Patent Owner's

constructions for these terms from the co-pending District Court litigation. (*See*

Petition, pp. 12–13; *See* Ex. 1010, pp. 3–12.) For the purposes of this proceeding,

Patent Owner agrees that the broadest reasonable interpretation of "management

program" is "separate tools or external agents having their own respective

command formats that provide management functions" and "command parse tree"

is "a hierarchical data representation having elements each specifying at least one

corresponding generic command component and a corresponding at least one

command action value."[1] (*See* Petition, pp. 12–13; Ex. 1010, pp. 3–12.)

> **B.   The Board should reject Petitioner's proposed construction for the term "recursively traversing" and instead adopt the plain and ordinary meaning.**

Patent Owner's Construction	Petitioner's Construction
Plain and ordinary meaning	traversing using a process that repeats itself

Petitioner proposed an additional construction for the term "recursively

traversing." As set forth below, the Board should reject Petitioner's construction

because it is inconsistent with the plain and ordinary meaning, as set forth in the

specification.

---

[1] Though the Petitioner suggests that it is proposing Patent Owner's

proposed construction for the term "command parse tree" from the district court,

Patent Owner's actual proposed construction ends with "... **command** action

value." Petitioner omits the word "command" from the proposed construction, but

Patent Owner contends the word "command" should be required for the

construction of the term "command parse tree," consistent with the explicit claim

language.

1.   **Petitioner's proposed construction for the term "recursively traversing" is incorrect and inconsistent with the plain and ordinary meaning of recursively traversing.**

The term "*recursively traversing*" appears in challenged dependent claims 3, 12, and 16: "recursively traversing the command parse tree based on an order of the input command words for identification of the matching token within the identified one element." Thus, each of these claims requires traversing the command parse tree, where the traversing is done recursively. ('526 patent, 9:44–48, 10:35–38, 11:3–7.) Petitioner does not construe the term traversing. Instead, Petitioner advances an incorrect construction for the term recursively—"using a process that repeats itself." (Petition, pp. 13–14.) The Board should reject Petitioner's incorrect construction, because, it is inconsistent with the specification and therefore cannot be the broadest reasonable interpretation. *Microsoft Corp. v. Proxyconn, Inc.*, 789 F.3d 1292, 1298 (Fed. Cir. 2015).

The '526 patent claims recite (and the specification describes) "[r]ecursively traversing the command parse tree based on an order of the input command words." A tree is a hierarchical structure comprised of sub-trees that themselves can be treated as trees. The claims require traversing the command parse tree in a recursive manner and is understood in the art as processing all nodes of a tree by systematically traversing each sub-tree of that tree until all sub-trees have been

completed, and traversing each sub-tree using the same procedure (by traversing each sub-tree of the sub-tree).

The patent explains what is meant by "recursively traversing" in the context of the command parse tree. When the command parse tree receives a generic input command, it searches for a tree element that includes a token value matching the first generic input command word. ('526 patent, 3:47–58, 4:7–12, Figure 3 step 42.) From there, starting at the children nodes descending from the matching tree element identified for the first generic input command word, the command parse tree is traversed and searched for tree elements that match subsequent generic input command words. ('526 patent, 3:55–58, 4:13–15, 4:19–27, Figure 3 step 48). This procedure continues for the entire generic input command received by the command parse tree, which results in a traversal of an entire branch of the command parse tree once evaluation of the complete generic input command occurs. ('526 patent, 4:27–36, Figure 3 step 48.) Thus, the patent describes that the recursive traversal of a command parse tree involves traversing the command parse tree and its sub-trees (descendant nodes of matching elements) while using the same validation procedure.

Petitioner's construction relies solely on Figure 3 of the '526 patent. From this figure, Petitioner speculates that "[b]ecause Figure 3 shows a process by which a set of steps is repeated in a looped manner, one of skill in the art would

understand that the broadest reasonable interpretation of 'recursively traversing'

includes such a looping mechanism." (Petition, pp. 13–14.) However, Petitioner

mischaracterizes Figure 3.

Figure 3 illustrates the **entire** process for "validation of generic commands,"

not just the recursive traversal of the command parse tree. (*See* '526 patent, 2:50–

52; Figure 3 step 42, "Traverse command Parse Tree" and step 48, "Traverse

Dependent Elements of Command Parse Tree.")
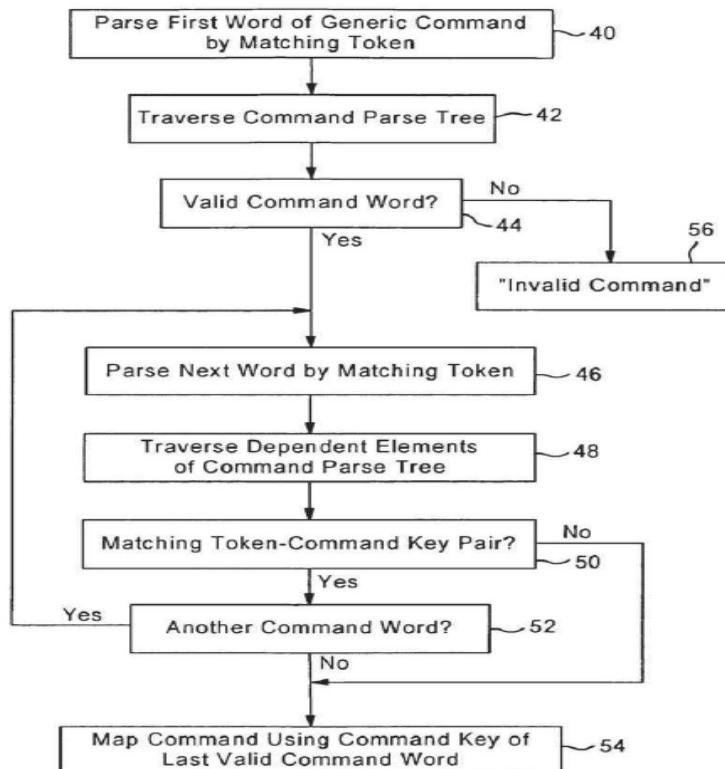
Figure 3

While Figure 3 shows that a portion of the validation process may be looped (*e.g.*,

step 52 loops back to step 46), it does not show that the recursive traversal of the

command parse tree includes merely a looping mechanism, as Petitioner contends.

- 13 -

Further, Petitioner's construction for recursively—"using a process that repeats itself"—in addition to being inconsistent with the specification, is impermissibly narrow. Neither the claims nor the specification require implementing this traversal using a specific technique, as Petitioner's construction requires. The term "recursive" only appears in the '526 patent when it is modifying the method of traversal. (*See* '526 patent, 3:55; claims 3, 12, 16, and 25.) The '526 patent does not require that the traversal must use a specific type of "recursion" or recursive functions. Thus, it would be improper for the Board to limit the *manner* of traversal (the path of travel) to a specific *implementation* of that traversal (*e.g.* use of a function that repeats itself). *See Silicon Graphics, Inc. v. ATI Technologies, Inc.*, 607 F.3d 784, 790–91 (Fed. Cir. 2010). Further, Petitioner's construction would exclude other types of recursive functionality that are well-known. (*See* Ex. 2001, Sanchez, p. 235 ("A common classification distinguishes the following types: linear (of which tail recursion is a special case), multiple (or exponential), nested, and mutual.)

Thus, for the purposes of this proceeding, the Board should reject the Petitioner's incorrect construction for this term, and instead rely on the plain and ordinary meaning. *See Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996) ("Words of a claim are generally given their ordinary and customary meaning.").

C.   Petitioner improperly limits the corresponding structure of the terms "means for validating"/"validating means."

Patent Owner's Construction	Petitioner's Construction
Function: "validating a generic command received from a user." Corresponding Structure: Parser 14 in Figure 2, which includes the command word translation table 20 and the command parse tree 22, as described in 3:36–61, and its equivalents.	Function: "validating a generic command received from a user." Corresponding Structure: Parser 14 of Figures 1 and 2, as described in 3:36–61, *and executing an algorithm as disclosed in Figure 3*.

Petitioner alleges that the corresponding structure for this term should include the '526 patent discussion of Figure 3 at 3:62–4:54. Petitioner is attempting to improperly limit the scope of the means-plus-function claim (beyond the function disclosed in the claims and the structure disclosed in the specification) to one method disclosed in an embodiment in the patent. That is contrary to the law. *Golight, Inc. v. Wal-Mart Stores, Inc.*, 355 F.3d 1327, 1335 (Fed. Cir. 2004) (finding no reason to import the preferred embodiment of a means plus function claim as a limitation during claim construction); *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1326 (Fed. Cir. 2002).

In contrast, Patent Owner cites the structure for the broader description of the validating process, as shown in Figure 2 and described in the specification. According to the '526 patent, Figure 2 discloses "in detail the parser … [which] includes a command word translation table 20 and a command parse tree 22 … [a] is configured for *validating a received generic command by comparing each input command word* to the command parse tree 22 to determine for the received generic command a tree element 24 identified as a best match." ('526 Pat., Ex. 1 at 3:36–51(emphasis added)). This portion of "[t]he specification … clearly links or associates [these] structure[s] to the [validating function] recited in the claim," and thus the Court should adopt Patent Owner's proposed structure. *Omega Eng'g, Inc., v. Raytek Corp.*, 334 F.3d 1314, 1321 (Fed. Cir. 2003).

IV.   Petitioner fails to meet its burden to establish that Martinez-Guerra renders obvious claims 1–26.

The Petition presents a single ground of unpatentability that Martinez-Guerra allegedly renders obvious claims 1–26. (Petition, pp. 19–20.) Rather than provide a detailed analysis of how Martinez-Guerra teaches or suggests each and every element of the claims, the Petition contains a series of cursory statements that the challenged claims are obvious. But, such a superficial analysis is insufficient to establish a *prima facie* case of obviousness. *KSR Int'l v. Teleflex Inc.*, 550 U.S. 398, 418 (2007) (an obviousness challenge must also provide some

"articulated reasoning with some rational underpinning to support the legal conclusion of obviousness").

As Patent Owner demonstrates below, when the details of Martinez-Guerra are fully considered, Petitioner's argument fails to establish that Martinez-Guerra discloses "management programs" or a command parse tree having elements "each specifying … a corresponding at least one command action value," a required element of each and every independent claim and their corresponding dependent claims. Martinez-Guerra's tools are not management programs because they are not separate or external from Martinez-Guerra's parser-translator component. And Petitioner relies solely on Martinez-Guerra's phrase structure rules, which are merely used for validation, as the claimed command parse tree. But Martinez-Guerra's phrase structure rules cannot be the claimed "command parse tree," because they do not contain "elements each specifying a command action value," as each of the independent claims requires.

A. An overview of Martinez-Guerra's parser-translator system.

Martinez-Guerra discloses a parser-translator system that receives a command in a first format and transforms the command into a second format. (Martinez-Guerra, Abstract, 4:56–59.) As explained by Martinez-Guerra, its system allows a user to "focus on the semantics of the desired operations and … not be concerned with the proper syntax of a language for a particular system.

(Martinez-Guerra, Abstract.) Although Martinez-Guerra seemingly provides an

outcome similar to that of the '526 patent, Martinez-Guerra's system uses different

techniques than the '526 patent.

FIG. 1
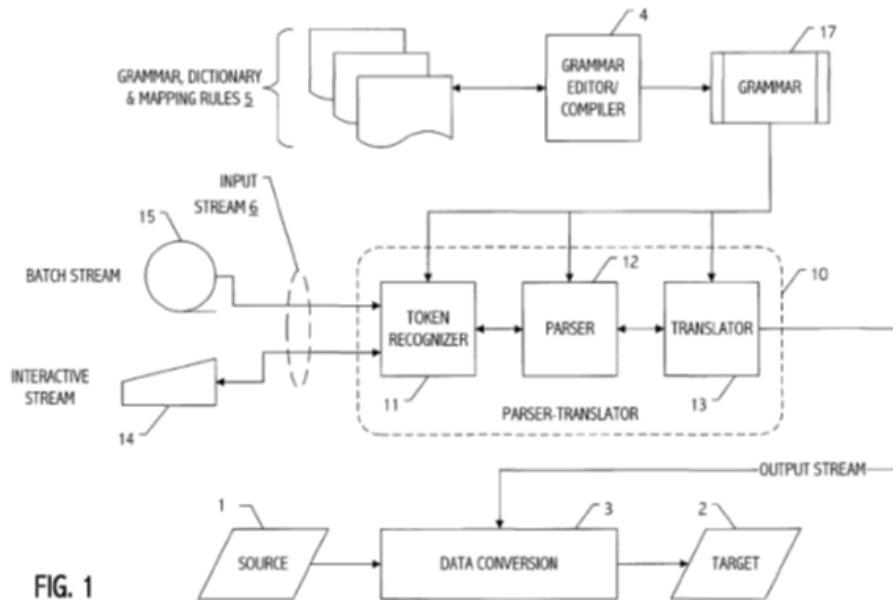
Martinez-Guerra discloses a parser-translator component that is directly integrated

into different tools. (Martinez-Guerra, 11:36–47.) Specifically, Martinez-Guerra's

parser-translator 10 includes three sub-components: token recognizer 11,

parser 12 and translator 13. Token recognizer 11 examines an input stream

command (*e.g.*, "delete pathname-expert") and determines if tokens are "legal"

based on grammar 17. (Martinez-Guerra, 10:42–45; 15:57.) Parser 12 accepts

tokens from token recognizer 11 and builds an internal representation of the

translation to be performed (*e.g.*, rm <pathname-expert>) (Martinez-Guerra,

10:49–53, 15:62.) Once it is determined that the internal representation is complete

(*i.e.*, phrase is complete), translator 13 translates the entire input stream. (Martinez-Guerra, 9:40–43.) The parser-translator then supplies a translated output stream that defines the particular data transformations to be performed, thereby transforming the command from a first format ("delete pathname-expert") to a second format (rm <pathname-expert>). (Martinez-Guerra, 9:45–62.)

The key to Martinez-Guerra's system is its grammar 17 that is defined as "a formal definition of the syntactic structure of a language typically represented as a set of rules that specify legal orderings of … phrases and symbols … in properly formed statements in a language." (Martinez-Guerra, 7:52–56.) Martinez-Guerra explains that the grammar 17 can be encoded with different parts. But, Martinez-Guerra provides no details how any of the parts of grammar 17 are arranged for use by its system. For example, Martinez-Guerra describes the components of a grammar encoding as:

> (1) a set of phrase structure rules and associated translation rules that define the syntactic order of a source and target language,
> (2) a dictionary of entries that describes each terminal category in the grammar and
> (3) identifier(s) for one or more user functions that may be called to establish parse context, obtain candidate next parse states and/or validate tokens, *e.g.*, by querying external stores.

(Martinez-Guerra, 7:56–64.)

In operation, upon receipt of a token in a first format, a phrase structure rule

of grammar 17 is applied to the token. (Martinez-Guerra, 10:42–45; 15:57.) After

the phrase structure rule is applied, the token is converted into the specified

terminal and nonterminal characters. The converted tokens are then translated into

a second format. (Martinez-Guerra, 9:45–62.) The mechanisms used by Martinez-

Guerra are nothing more than conventional parsing and translation techniques.

B.   Petitioner fails to establish that Martinez-Guerra discloses "executing a plurality of management programs, according to respective command formats," as required by claims 1, 10, 14, and 23.

The preamble for each of the independent claims recites "executing a

plurality of management programs.…" (*See e.g.,* claim 1 (9:20–21); claim 10

(10:11–13); claim 14 (10:44–46); and claim 23 (12:1–3). In this case, the preamble

of each independent claim should be afforded patentable weight, because the

recited "management programs" are necessary to give meaning to the claims of the

'526 patent. *Catalina Marketing Int'l v. Coolsavings.com, Inc.*, 289 F.3d 801, 808

(Fed. Cir. 2002) ("In general, a preamble limits the invention if it recites essential

structure or steps, or if it is 'necessary to give life, meaning, and vitality' to the

claim."). Here, the claimed "management programs" are inseparably linked to the

claimed system and define the claims, because without "management programs,"

the claims would not be able to perform additional claim elements, such as

"issuing a prescribed command of a selected one of the management programs."

(*See* '526 patent, 9:32–34, 10:23–24, 10:58–59, 12:15–16.)

Petitioner fails to establish that Martinez-Guerra teaches the subject matter of the preamble. Specifically, Martinez-Guerra does not disclose a system for executing a plurality of *management programs* because the multiple instances of Martinez-Guerra's parser-translator component are integrated directly into Martinez-Guerra's different tools. (Martinez-Guerra, 11:36–47.) And as discussed above in Section III.A, both parties agree that "management programs" should be construed to mean "*separate* tools or *external* agents having their own respective command formats that provide management functions." Because the parser-translator component is integrated into the different tools, Martinez-Guerra's tools are not "separate tools or external agents," as required by the claimed management programs.

Martinez-Guerra discloses integrating its parser-component directly into different "enterprise tools"—an implementation that is different from the '526 patent's implementation where the parser and management programs are separate. (Martinez-Guerra, 11:36.) For example, Martinez-Guerra explains that an "enterprise tools environment including data discovery and cleansing tools, data extraction conversion and migration tools, data movement and replication tools, query Multi-Dimensional Data (MDD) analysis and On-Line Analytical Processing

(OLAP) tools, as well as applications and gateways … may advantageously

incorporate a parser-translator component." (Martinez-Guerra, 11:36–42.) Thus,

Martinez-Guerra's parser-translator component and grammars are incorporated

directly into a "multiplicity of tools" and these tools are different from the claimed

"management programs": "*separate* tools or *external* agents having their own

respective command formats that provide management functions." (*See* Martinez-

Guerra, 11:42–47; Section III.A above.)

In contrast, consistent with the claims, the '526 patent describes a system

where its management programs are separate and/or external from its parser

component. (*See* Section III.A.) Figure 1 of the '526 patent (reproduced below)

illustrates an embodiment having management programs 18a-18d separate and/or

external from the parser 14. (*See* also '526 patent, 3:1–15.)
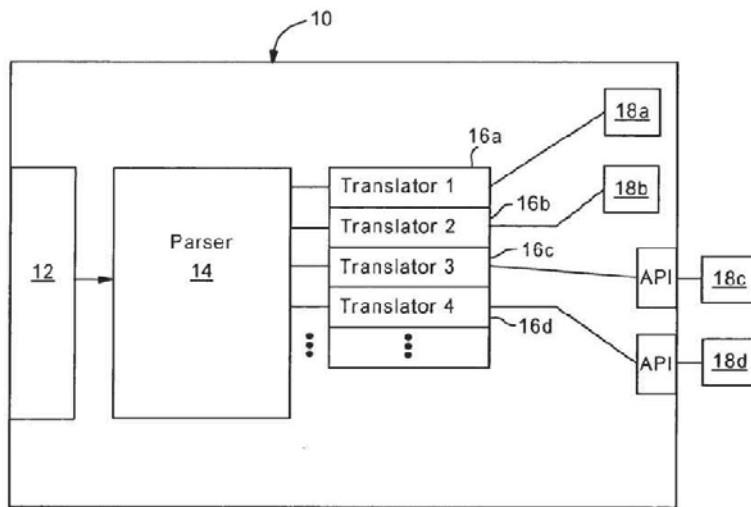

Figure 1

As highlighted by Figure 1, instead of incorporating the parser component into each of the management programs, the '526 patent "incorporates the functionality of all external administrative [tools]" into the parser component, thereby allowing for a single tool that controls multiple management programs. ('526 patent, 3:21–27, Abstract.) Because Martinez-Guerra integrates its parser-component into each of its tools, Martinez-Guerra does not provide a similar single tool that controls multiple separate and/or external administrative tools. Therefore, Martinez-Guerra does not disclose "executing a plurality of *management programs ...*," as required by the claims.

Accordingly, Petitioner fails to meet its burden to establish that Martinez-Guerra renders obvious independent claims 1, 10, 14, and 23, and the Board should deny institution for all claims.

C.   Petitioner fails to establish that Martinez-Guerra discloses or renders obvious a "command parse tree" as required by claims 1, 10, 14, and 23.

Each of the independent claims further requires validating a generic command based on a command parse tree. (*See e.g.,* claim 1 (9:23–25); claim 10 (10:14–18); claim 14 (10:49–51); and claim 23 (12:4–7)[2].) In turn, each of these

---

[2] While claim 23 does not explicitly recite a command parse tree, claim 23 recites "means for validating a generic command received from a user, the validating means configured for specifying valid generic commands … and having

- 23 -

independent claims specifies the structure of the command parse tree. Specifically, the command parse tree has "elements each specifying at least one corresponding generic command component and a corresponding at least one command action value." Petitioner fails to meet its burden to show obviousness of independent claims 1, 10, 14, and 23 because what Petitioner alleges is the command parse tree—Martinez-Guerra's phrase structure rule—does not include a corresponding at least one command action value.

---

elements each specifying at least one generic command component and a corresponding at least one command action value." ('526 patent, 12:4–10.) The means for validating reads on the disclosed structure, namely the disclosed parser 14 of Figure 2 which includes the command parse tree. Thus, any discussion regarding the command parse tree with respect to independent claims 1, 10, and 14 applies equally to independent claim 23.

1. The phrase structure rules of Martinez-Guerra have a different format than the claimed command parse tree.

Martinez-Guerra's system uses a grammar to perform its language translations. (Martinez-Guerra, 7:52–64, 16:6–24.) Martinez-Guerra's grammar is "typically represented as a set of rules that specify legal orderings of constituents and subconstituents (e.g., phrases and symbols) in properly formed statements (or sentences) in a language." (Martinez-Guerra, 7:53–56.) The phrase structure rules relied on by Petitioner are a set of rules for Martinez-Guerra's grammar. The phrase structure rules include two parts: the name of the rule and the rule specification. (Martinez-Guerra, 14:33–34.) Below is an example of a rule (*e.g.*, phrase structure rule) from Martinez-Guerra:

$$A \rightarrow B \ c \ D$$

(Martinez-Guerra, 14:36.) In this example, the name of the rule (left side) is "A" and the rule specification (right side) is "B c D." (Martinez-Guerra, 14:37–39, 14:45–53.) The rule specification indicates the legal sequence of tokens allowed by the programming language. (Martinez-Guerra, 14:55–57, 8:34–35.) Thus, if rule A is applied to an input token, it will be expanded to B c D. (Martinez-Guerra, 14:42.) That is, if C -> A e Y, the result of the expansion would be C -> (B c D) e Y because "A" = "B c D."

Petitioner contends that Martinez-Guerra's phrase structure rules are the claimed command parse tree, arguing that the "phrase structure rules encoded in

- 25 -

the grammar … constitute a command parse tree, because they are collectively a

hierarchical data representation of the valid component of statements in a high-

level language." (Petition, p. 23 (emphasis added).) But, being a hierarchical data

representation is not enough. Petitioner must establish that the phrase structure

rules have the same structure and format as the claimed command parse tree.

Petitioner fails to make this showing because the format of Martinez-Guerra's

phrase structure tree is different. Specifically, as discussed in detail below,

nowhere does Martinez-Guerra disclose or suggest that its phrase structure rules

specify "*at least one command action value*," as required by each of the challenged

claims. ('526 patent, 9:23–29, 10:15–21, 10:5–5, 12:4–10 (emphasis added).)

> **2.** **Martinez-Guerra's phrase structure rules are not arranged in a tree having elements that specify a corresponding at least one command action value.**

Petitioner's argument fails for a fundamental reason—Petitioner does not

establish that the phrase structure rule (alleged command parse tree) has elements

that specify a corresponding at least one command action value. Petitioner points

to the rule specification portion of the phrase structure rules as the alleged generic

command component of the command parse tree element. (Petition, p. 25.)

Petitioner does not and cannot point to another portion of the phrase structure rules

for the claimed corresponding command action value because no additional parts

of Martinez-Guerra's phrase structure rule exists, leaving Petitioner at a dead end.

Faced with this predicament, Petitioner attempts to further contort the teachings of Martinez-Guerra, by arguing that Martinez-Guerra's "instruction to return an error that results from inputting an invalid word or incomplete sequence" or that Martinez-Guerra's "translation function that results from inputting a complete and valid sequence of tokens" equal the claimed "corresponding at least one command action value" of an element of the command parse tree. (Petition, p. 26.) These arguments are fatally flawed.

Martinez-Guerra's error instruction does not constitute a command action value. Martinez-Guerra provides only a minimal disclosure regarding its error reporting mechanisms: "if an input is invalid, an error may be reported via the user interface or by other mechanism such as a log file." (Martinez-Guerra, 19:3–4.) The analysis that determines whether an input is invalid (*i.e.*, the error triggering event) occurs separately from Martinez-Guerra's processing using its phrase structure rules. Martinez-Guerra's phrase structure rules only take into account the *legal* sequences of tokens allowed by the language. (Martinez-Guerra, 8:24–25.) As a consequence, Martinez-Guerra's system does not associate any error reported with its phrase structure rule processing. Indeed, Martinez-Guerra never establishes any link between its error reporting and phrase structure rules. Accordingly, Martinez-Guerra's phrase structure rules (the alleged command parse tree) do not

include the error reporting (the alleged command action value), as required by the claims.

Martinez-Guerra's translation function also does not constitute a command action value. Petitioner alleges that the "translation function that results from inputting a complete and valid sequence of token … is what the '526 patent refers to as a "command action value." (Petition, p. 26.) This is not true, because Martinez-Guerra's translation function is not a command action value that is part of an element of a command parse tree. First, a translation function is not a part of a phrase structure rule (the alleged command parse tree) which has only two parts: the name of the rule and the rule specification (*e.g.*, A→B c D). Petitioner provides no evidence to the contrary. Second, Martinez-Guerra explicitly states that the translation function "specifies *how the sequence of input tokens matching the rule expansion* should be translated, or mapped." (Martinez-Guerra, 15:25–27 (emphasis added).) According to Martinez-Guerra, a translation function only applies to a *complete* valid sequence of tokens. (Martinez-Guerra, 9:40–43.) Thus, in contrast to the '526 patent, each token ingested by Martinez-Guerra's parser does not have a corresponding translation function. Instead, Martinez-Guerra's translation functions are only applied once Martinez-Guerra's system determines that it has received the entirety of a complete valid sequence of tokens. (Martinez-Guerra, 9:40–43; 15:25–27.)

3. Conclusion

Petitioner points to no disclosure in Martinez-Guerra teaching or suggesting

the specific format required by the claimed command parse tree: "command parse

tree … having elements each specifying at least one corresponding generic

command component and a corresponding at least one command action value."

Therefore, Petitioner failed to establish that Martinez-Guerra renders any of the

challenged claims obvious and the Board should accordingly deny institution of

Ground 1.

> **D.   Martinez-Guerra does not disclose the first requirement of claim 2: "comparing each input command word to a command word translation table, configured for storing for each prescribed command word a corresponding token, for identification of a matching token."**

Petitioner contends that Martinez-Guerra's "dictionary entries comprise a

command word translation table used by the token recognizer to match input words

with valid tokens in the phrase structure rules." (Petition, p. 29.) But Martinez-

Guerra's dictionary entries differ from the claimed command word translation

table, because the dictionary entries do not store both tokens and command words.

Claim 2 requires that "command word translation table" store "for each

prescribed command word" a "**corresponding token**." That is, the "command

word translation table" requires the storage of both "command words" and

"tokens" to allow for "identification of a matching token" to an "input command

- 29 -

word." (Claim 2.) This understanding of claim 2 is also consistent with the '526

patent specification. For example, consider Figure 2 (reproduced below).

Word	Token	
connection	1	
connections	2	← 20
get	3	
info	4	
set	5	
tcp	6	
udp	7	
watch	8	
< Invalid Toket>	-1	

    26   28

As illustrated in Figure 2, command word translation table 20 includes a set of

prescribed command words 26 and token values 28. Each command word has a

corresponding token. For example, the prescribed command word "watch" has a

corresponding token value of 8. ('526 patent, 4:8–10.)

Martinez-Guerra's dictionary cannot be the claimed "command word

translation table" because Martinez-Guerra's dictionary only stores what Martinez-

Guerra refers to as a "token." Specifically, Martinez-Guerra discloses that the

entries in its dictionary are "all the tokens that a parser-translator (e.g., token

recognizer 31 of parser-translator component 30) should recognize." (Martinez-

Guerra, 14:17–19.) That is, at most, Martinez-Guerra's dictionary entries include a

list of all tokens that should be recognized by Martinez-Guerra's system as legal

tokens. (Martinez-Guerra, 7:36–37, 14:17–19.) But, even if one could equate

Martinez-Guerra's "token" with the claimed "token" (which Patent Owner does

not concede), Petitioner still has not pointed to any other disclosure of Martinez-Guerra that teaches or even suggests that the dictionary also stores corresponding "command words" for each "token." This is not surprising because, as explained above, Martinez-Guerra's dictionary does not store anything other than what it refers to as "tokens."

Accordingly, Petitioner failed to meet its burden to establish that Martinez-Guerra renders obvious claims 2, 11, and 15 and the Board should deny institution as to dependent claims 2, 11 and 15 for this further reason.

E.   Petitioner fails to establish that Martinez-Guerra discloses or renders obvious claims 3, 12, 16, and 25.

Each of the dependent claims 2, 11, 15, and 24 require determining a presence of the matching token within the command parse tree for each input command word. (*See* '526 patent, 9:42–43, 10:32–34, 11:1–2, 12:25–26.) Dependent claims 3, 12, 16, and 25 further narrow the "determining" element by requiring "recursively traversing the command parse tree based on an order of the input command words for identification of the matching token within the identified one element." Martinez-Guerra cannot render these claims obvious because it does not disclose "recursively traversing the command parse tree," either under Petitioner's incorrect claim construction or the plain and ordinary meaning of "recursively traversing." (*See* Section. III.B above.)

1. Petitioner fails to show that Martinez-Guerra discloses recursively traversing a command parse tree under its own flawed construction.

Petitioner argues an incorrect construction of "recursively traversing" to mean "traversing using a process that repeats itself." (Petition, p. 13.) Patent Owner, in Section III.B, showed the flaws with this construction. Nonetheless, even if the Board adopts Petitioner's construction, Petitioner still fails to demonstrate that Martinez-Guerra discloses recursively traversing a command parse tree. Petitioner's bare-bones argument focuses solely on Martinez-Guerra's discussion regarding parse states, but Petitioner never explains how this disclosure constitutes recursively traversing a command parse tree. (*See* Petition, pp. 32–33.) And Martinez-Guerra's minimalist discussion about updating parse states make no mention of a recursive traversal of Martinez-Guerra's phrase structure rules—the alleged command parse tree.

Martinez-Guerra simply discloses that its parser receives a set of input tokens and applies each of the tokens to parse states maintained by the parser. (Martinez-Guerra, 9:34–44.) Martinez-Guerra's parse states are capable of "maintain[ing] a representation of tokens selected so far, of a current position within the grammar rules, and of the translations of tokens and/or phrase structure rules completed so far." (Martinez-Guerra, 9:63–66.) According to Martinez-Guerra, the input tokens are applied to each of the parse states, starting with a first

parse state, and if a given parse state looks for the applied input token, that parse state is updated by adding the token to the parse state. (Martinez-Guerra, 9:40–58.) If the addition of a token completes the governing phrase structure rule (*i.e.*, a complete phrase has been parsed), Martinez-Guerra's translator is invoked and translates the string of tokens. (Martinez-Guerra, 9:45–48.)

Thus, while Martinez-Guerra discloses the updating of parse states for each parse token, Martinez-Guerra never explains that the parse state updating requires recursively traversing (or "traversing using a process that repeats itself," according to Petitioner's construction) elements of Martinez-Guerra's phrase structure rules, the alleged claimed command parse tree. And because of this foundational flaw, Petitioner cannot point to a recursive traversal of the phrase structure rules.

Accordingly, Petitioner failed to meet its burden to establish that Martinez-Guerra renders obvious claims 3, 12, and 16 under its flawed claim construction.

2. Petitioner failed to establish that Martinez-Guerra discloses recursively traversing a command parse tree under the plain and ordinary meaning.

Petitioner, perhaps recognizing the flaws in its proposed construction for "recursively traversing," contends that "even if the Board were to construe this claim limitation in accordance with the mathematical meaning of 'recursive'—i. *e.,* 'a function that calls itself'—Martinez-Guerra still renders this limitation obvious." (Petitioner, p. 33.) Specifically, Petitioner states that "[s]killed artisans understood

at the time of the alleged invention of the '526 patent that repeating processes and recursive processes were known alternatives for performing tasks such as those claimed, and that a recursive process could be substituted for a repeating one with predictable results." (Petition, pp. 33–34.) Petitioner's bold pronouncement ignores the plain and ordinary meaning of "recursive" and the explicit language of the claims that require traversing a command parse tree.

As discussed above in Section III.B, the '526 patent demonstrates what it means by "recursively traversing a command parse tree." Specifically, the '526 patent describes searching for each generic input command word and using each matched tree element as the root for subsequent searches for input command words. ('526 patent, 3:47–58, 4:7–12, 3:55–58, 4:13–15, 4:19–27, 4:27–36, Figure 3 steps 42 and 48.) Thus, the recursive traversal of a command parse tree involves traversing the command parse tree and its sub-trees (descendant nodes of matching tree elements) while using the same validation procedure. Martinez-Guerra does not disclose such a recursive traversal of a command parse tree. Petitioner merely points to Martinez-Guerra's parse state method, which repeats for each input token. (*See* Petition, pp. 33–34.) But this is not enough.

Accordingly, Petitioner failed to meet its burden to establish that Martinez-Guerra renders obvious claims 3, 12, 16 and 25 and the Board should deny institution as to dependent claims 3, 12, 16 and 25 for this further reason.

IPR2016-00119
U.S. Pat. No. 7,047,526

V.   Conclusion.

Petitioner has not met its burden to prove that claims 1–26 are unpatentable. Petitioner relies on a single reference, Martinez-Guerra, to assert its obviousness challenge. But Patent Owner has fully met that challenge, carefully and repeatedly cataloging Petitioner's failures to meet its burden and showing that Petitioner has not established that the cited reference renders obvious any of the challenged claims of the '526 patent. Accordingly, the Board should find claims 1–26 patentable over the proposed grounds and deny Institution.

Respectfully submitted,

STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.

Lori A. Gordon (Reg. No. 50,633)
Attorney for Patent Owner

Date: February 18, 2016
1100 New York Avenue, N.W.
Washington, D.C. 20005
(202) 371-2600

IPR2016-00119
U.S. Pat. No. 7,047,526

CERTIFICATION OF SERVICE

The undersigned hereby certifies that the foregoing **PATENT OWNER**

PRELIMINARY RESPONSE and all associated exhibits were served

electronically via e-mail on February 18th, 2016, in their entirety on the following:

Eugene M. Paige (Lead Counsel)
Robert A. Van Nest (Back-up Counsel, pending filing of *pro hac vice* motion)
Brian L. Ferrall (Back-up Counsel, pending filing of *pro hac vice* motion)
David J. Silbert (Back-up Counsel, pending filing of *pro hac vice* motion)
KEKER & VAN NEST LLP
aristaipr@kvn.com
epaige@kvn.com
dsilbert@kvn.com

STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.

Lori A. Gordon (Reg. No. 50,633)
Attorney for Patent Owner

Date: February 18, 2016

1100 New York Avenue, N.W.
Washington, D.C.20005
(202) 371-2600